

1 **REDUCED-ERROR PROCESSING OF TRANSFORMED DIGITAL DATA**2 **CROSS REFERENCES**

3 The present application is related to the following
4 applications even dated herewith: Attorney docket number
5 Y0998-372, entitled, "Transform-domain correction of real-
6 domain errors," by inventors J. Mitchell et al., and
7 Attorney docket number Y0998-373, entitled, "Error reduction
8 in transformed digital data," by inventors M. Bright et al.,
9 which are incorporated herein in entirety by reference.

10 **FIELD OF THE INVENTION**

11 This invention relates to transform coding of digital
12 data, specifically to real domain processing of transform
13 data. More particularly, this invention relates to
14 reduced-error digital processing of inverse transformed
15 data.

16 **BACKGROUND OF THE INVENTION**

17 Transform coding is the name given to a wide family of
18 techniques for data coding, in which each block of data to
19 be coded is transformed by some mathematical function prior
20 to further processing. A block of data may be a part of a
21 data object being coded, or may be the entire object. The

1 data generally represent some phenomenon, which may be for
2 example a spectral or spectrum analysis, an image, an audio
3 clip, a video clip, etc. The transform function is usually
4 chosen to reflect some quality of the phenomenon being
5 coded; for example, in coding of audio, still images and
6 motion pictures, the Fourier transform or Discrete Cosine
7 Transform (DCT) can be used to analyze the data into
8 frequency terms or coefficients. Given the phenomenon being
9 coded, there is generally a concentration of the information
10 into a few frequency coefficients. Therefore, the
11 transformed data can often be more economically encoded or
12 compressed than the original data. This means that
13 transform coding can be used to compress certain types of
14 data to minimize storage space or transmission time over a
15 communication link.

16 An example of transform coding in use is found in the
17 Joint Photographic Experts Group (JPEG) international
18 standard for still image compression, as defined by *ITU-T*
19 *Rec. T.81 (1992) | ISO/IEC 10918-1:1994, Information*
20 *technology — Digital compression and coding of*
21 *continuous-tone still images, Part 1: Requirements and*
22 *Guidelines*. Another example is the Moving Pictures Experts
23 Group (MPEG) international standard for motion picture
24 compression, defined by *ISO/IEC 11172:1993, Information*
25 *Technology — Coding of moving pictures and associated audio*
26 *for digital storage media at up to about 1,5 Mbits/s*. This
27 MPEG-1 standard defines systems for both video compression
28 (Part 2 of the standard) and audio compression (Part 3). A
29 more recent MPEG video standard (MPEG-2) is defined by *ITU-T*
30 *Rec. H.262 | ISO/IEC 13818-2: 1996 Information Technology —*

1 Generic Coding of moving pictures and associated audio --
2 Part 2: video. A newer audio standard is ISO/IEC 13818-3:
3 1996 Information Technology — Generic Coding of moving
4 pictures and associated audio -- Part 3: audio. All three
5 image international data compression standards use the DCT
6 on 8x8 blocks of samples to achieve image compression. DCT
7 compression of images is used herein to give illustrations
8 of the general concepts put forward below; a complete
9 explanation can be found in Chapter 4 "The Discrete Cosine
10 Transform (DCT)" in W. B. Pennebaker and J. L. Mitchell,
11 *JPEG: Still Image Data Compression Standard*, Van Nostrand
12 Reinhold: New York, (1993).

13 Wavelet coding is another form of transform coding.
14 Special localized basis functions allow wavelet coding to
15 preserve edges and small details. For compression the
16 transformed data is usually quantized. Wavelet coding is
17 used for fingerprint identification by the FBI. Wavelet
18 coding is a subset of the more general subband coding
19 technique. Subband coding uses filter banks to decompose the
20 data into particular bands. Compression is achieved by
21 quantizing the lower frequency bands more finely than the
22 higher frequency bands while sampling the lower frequency
23 bands more coarsely than the higher frequency bands. A
24 summary of wavelet, DCT, and other transform coding is given
25 in Chapter 5 "Compression Algorithms for Diffuse Data" in
26 Roy Hoffman, *Data Compression in Digital Systems*, Chapman
27 and Hall: New York, (1997).

28 In any technology and for any phenomenon represented by
29 digital data, the data before a transformation is performed
30 are referred to as being "in the real domain". After a

1 transformation is performed, the new data are often called
2 "transform data" or "transform coefficients", and referred
3 to as being "in the transform domain". The function used to
4 take data from the real domain to the transform domain is
5 called the "forward transform". The mathematical inverse of
6 the forward transform, which takes data from the transform
7 domain to the real domain, is called the respective "inverse
8 transform".

9 In general, the forward transform will produce
10 real-valued data, not necessarily integers. To achieve data
11 compression, the transform coefficients are converted to
12 integers by the process of quantization. Suppose that (λ_i)
13 is a set of real-valued transform coefficients resulting
14 from the forward transform of one unit of data. Note that
15 one unit of data may be a one-dimensional or two-dimensional
16 block of data samples or even the entire data. The
17 "quantization values" (q_i) are parameters to the encoding
18 process. The "quantized transform coefficients" or
19 "transform-coded data" are the sequence of values (a_i)
20 defined by the quantization function Q :

$$21 \quad a_i = Q(\lambda_i) = \left\lfloor \frac{\lambda_i}{q_i} + 0.5 \right\rfloor \quad (1)$$

22 where $\lfloor x \rfloor$ means the greatest integer less than or equal to x .

23 The resulting integers are then passed on for possible
24 further encoding or compression before being stored or
25 transmitted. To decode the data, the quantized coefficients
26 are multiplied by the quantization values to give new
27 "dequantized coefficients" (λ'_i) given by

$$28 \quad \lambda'_i = q_i a_i. \quad (2)$$

1 The process of quantization followed by dequantization
2 (also called inverse quantization) can thus be described as
3 "rounding to the nearest multiple of q_i ". The quantization
4 values are chosen so that the loss of information in the
5 quantization step is within some specified bound. For
6 example, for audio or image data, one quantization level is
7 usually the smallest change in data that can be perceived.
8 It is quantization that allows transform coding to achieve
9 good data compression ratios. A good choice of transform
10 allows quantization values to be chosen which will
11 significantly cut down the amount of data to be encoded.
12 For example, the DCT is chosen for image compression because
13 the frequency components which result produce almost
14 independent responses from the human visual system. This
15 means that the coefficients relating to those components to
16 which the visual system is less sensitive, namely the
17 high-frequency components, may be quantized using large
18 quantization values without perceptible loss of image
19 quality. Coefficients relating to components to which the
20 visual system is more sensitive, namely the low-frequency
21 components, are quantized using smaller quantization values.

22 The inverse transform also generally produces
23 non-integer data. Usually the decoded data are required to
24 be in integer form. For example, systems for the playback
25 of audio data or the display of image data generally accept
26 input in the form of integers. For this reason, a transform
27 decoder generally includes a step that converts the
28 non-integer data from the inverse transform to integer data,
29 either by truncation or by rounding to the nearest integer.
30 There is also often a limit on the range of the integer data

1 output from the decoding process in order that the data may
2 be stored in a given number of bits. For this reason the
3 decoder also often includes a "clipping" stage that ensures
4 that the output data are in an acceptable range. If the
5 acceptable range is $[a,b]$, then all values less than a are
6 changed to a , and all values greater than b are changed to
7 b .

8 These rounding and clipping processes are often
9 considered an integral part of the decoder, and it is these
10 which are the cause of inaccuracies in decoded data and in
11 particular when decoded data are re-encoded. For example,
12 the JPEG standard (Part 1) specifies that a source image
13 sample is defined as an integer with precision P bits, with
14 any value in the range 0 to $2^P - 1$. The decoder is
15 expected to reconstruct the output from the inverse discrete
16 cosine transform (IDCT) to the specified precision. For the
17 baseline JPEG coding P is defined to be 8; for other
18 DCT-based coding P can be 8 or 12. The MPEG-2 video
19 standard states in Annex A (Discrete cosine transform) "The
20 input to the forward transform and the output from the
21 inverse transform is represented with 9 bits."

22 For JPEG the compliance test data for the encoder
23 source image test data and the decoder reference test data
24 are 8 bit/sample integers. Even though rounding to integers
25 is typical, some programming languages convert from floating
26 point to integers by truncation. Implementations in
27 software that accept this conversion to integers by
28 truncation introduce larger errors into the real-domain
29 integer output from the inverse transform.

1 The term "high-precision" is used herein to refer to
2 numerical values which are stored to a precision more
3 accurate than the precision used when storing the values as
4 integers. Examples of high-precision numbers are
5 floating-point or fixed-point representations of numbers.

6 SUMMARY OF THE INVENTION

7 In light of the problems described above regarding
8 inaccuracies caused by digital processing techniques and by
9 such things as rounding and clipping after the inverse
10 transform of transform data, one aspect of this invention
11 provides a method for processing transform data in the real
12 domain. This method reduces the undesired errors in the
13 data produced by such things as rounding to integers and
14 clipping to an allowed range after the inverse transform.
15 In an embodiment, this method includes: performing the
16 inverse transform of the transform data such that the
17 real-domain data produced are in the form of high-precision
18 numbers; processing these high-precision numbers; and
19 converting the processed high-precision numbers to integers
20 and clipping to an allowed range only after the processing
21 stage is complete.

22 It is another aspect of this invention to provide a
23 method for processing transform-coded data in the real
24 domain which reduces the undesired errors in the data
25 produced by the converting to integers and clipping to an
26 allowed range after the inverse transform. In an
27 embodiment, the method includes: performing the inverse

1 quantization of the transform-coded data; performing the
2 inverse transform of the transform data thus produced, such
3 that the real-domain data produced are in the form of
4 high-precision numbers; processing these high-precision
5 numbers; and converting the processed high-precision numbers
6 to integers and clipping to an allowed range only after the
7 processing stage is complete.

8 Still another aspect of the present invention is to
9 provide a method for processing transform-coded data in the
10 real domain to produce new transform-coded data, which
11 reduces the error produced by converting to integers and
12 clipping to an allowed range after the inverse transform.
13 In an embodiment, this method includes: performing the
14 inverse quantization of the transform-coded data; performing
15 the inverse transform of the transform data thus produced,
16 such that the real-domain data produced are in the form of
17 high-precision numbers; processing these high-precision
18 numbers; performing the forward transform on the processed
19 high-precision numbers; and performing quantization on the
20 new transform data. If the errors in the forward and
21 inverse transforms and in the processing are sufficiently
22 small, there will be no undesirable errors produced in the
23 new quantized transform-domain data.

24 There is no requirement that the input data to the
25 methods described herein need come from a single data
26 source. Thus, this invention is not restricted to the
27 real-domain processing of data from a single source, but
28 also applies to real-domain processing of data from multiple
29 sources, such as the merging of images or audio data.

1 The quantization described in the background is the
2 linear quantization used in international image data
3 compression standards such as JPEG and MPEG. There is no
4 requirement that the quantization be linear. Any mapping
5 that reduces the number of transform data levels in a
6 deterministic way can be used with this invention. The
7 quantization step has been described mathematically with a
8 division in Equation (1). Actual embodiments may use a
9 lookup table or a sequence of comparisons to achieve similar
10 results.

11 It is a further aspect of the invention to provide
12 apparatus, a computer product and an article of manufacture
13 comprising a computer usable medium having computer readable
14 program code means embodied therein for causing a computer
15 to perform the methods of the present invention.

16 **BRIEF DESCRIPTION OF FIGURES**

17 These and other objects, features, and advantages of
18 the present invention will become apparent upon further
19 consideration of the following detailed description of the
20 invention when read in conjunction with the drawing figures,
21 in which:

22 FIG. 1(a) is a block diagram showing a method for
23 performing an inverse transform;

24 FIG. 1(b) is a block diagram showing a system for
25 performing an inverse transform;

26 FIG. 2(a) is a block diagram showing a method for
27 decoding transform-coded data;

1 FIG. 2(b) is a block diagram showing a system for
2 decoding transform-coded data;

3 FIG. 3 is a block diagram showing a method for the
4 real-domain processing of transform data;

5 FIG. 4 is a block diagram showing a method for
6 performing an inverse transform followed by a forward
7 transform, and demonstrating the multi-generation problem;

8 FIG. 5 is a block diagram showing a method for decoding
9 and re-encoding transform-coded data, and demonstrating the
10 multi-generation problem;

11 FIG. 6 is a block diagram showing a method for
12 performing an inverse transform, real-domain data
13 manipulation and a forward transform, and demonstrating the
14 multi-generation problem;

15 FIG. 7(a) is a block diagram showing a method for
16 performing real-domain processing of JPEG DCT-coded image
17 data, which exhibits the multi-generation problem;

18 FIG. 7(b) is a block diagram showing a system for
19 performing real-domain processing of JPEG DCT-coded image
20 data, which exhibits the multi-generation problem;

21 Fig. 8(a) gives the JPEG example luminance quantization
22 matrix;

23 Fig. 8(b) gives the JPEG example chrominance
24 quantization matrix;

25 FIG. 8(c) is a numerical example of how real-domain
26 rounding can cause significant errors in 8x8 block DCT coded
27 data;

1 FIG. 8(d) is a numerical example of how real-domain
2 truncation can cause significant errors in 8x8 block DCT
3 coded data;

4 FIG. 8(e) is a series of graphs illustrating how
5 real-domain clipping can cause errors in one-dimensional
6 discrete cosine transform-coded data;

7 FIG. 8(f) and FIG. 8(g) are a numerical example of how
8 real-domain clipping can cause significant errors in 8x8
9 block DCT coded data;

10 FIG. 9 is a block diagram showing a method performing
11 multiple iterations of the process described in FIG. 5, and
12 exhibiting the multi-generation problem;

13 FIG. 10 is a block diagram showing a method for
14 performing multiple iterations of real-domain manipulations,
15 and exhibiting the multi-generation problem;

16 FIG. 11(a) is a block diagram showing an example of a
17 method for reduced-error processing of transform data in
18 accordance with the present invention;

19 FIG. 11(b) is a block diagram showing an example of a
20 system for reduced-error processing of transform data in
21 accordance with the present invention;

22 FIG. 12(a) is a block diagram showing an example of a
23 method for performing an inverse transform followed by a
24 forward transform, such that this process is lossless in
25 accordance with the present invention;

26 FIG. 12(b) is a block diagram showing an example of a
27 system for performing an inverse transform followed by a

1 forward transform, such that this process is lossless in
2 accordance with the present invention;

3 FIG. 13(a) is a block diagram showing an example of a
4 method for performing real-domain manipulation of transform
5 data with reduced error followed by a forward transform in
6 accordance with the present invention;

7 FIG. 13(b) is a block diagram showing an example of a
8 system for performing real-domain manipulation of transform
9 data with reduced error followed by a forward transform in
10 accordance with the present invention;

11 FIG. 14(a) is a block diagram showing an example of a
12 method for reduced-error processing of transform-coded data
13 in accordance with the present invention;

14 FIG. 14(b) is a block diagram showing an example of a
15 system for reduced-error processing of transform-coded data
16 in accordance with the present invention;

17 FIG. 15(a) is a block diagram showing an example of a
18 method for decoding and re-encoding transform-coded data
19 such that this process is lossless in accordance with the
20 present invention;

21 FIG. 15(b) is a block diagram showing an example of a
22 system for decoding and re-encoding transform-coded data
23 such that this process is lossless in accordance with the
24 present invention;

25 FIG. 16(a) is a block diagram showing an example of a
26 method for performing real-domain manipulation of
27 transform-coded data with reduced error in accordance with
28 the present invention;

1 FIG. 16(b) is a block diagram showing an example of a
2 system for performing real-domain manipulation of
3 transform-coded data with reduced error in accordance with
4 the present invention;

5 FIG. 17(a) is a block diagram showing an example
6 embodiment of a method for performing real-domain processing
7 of JPEG-coded image data, such that undesired errors in the
8 new transform-coded data are reduced or eliminated in
9 accordance with the present invention;

10 FIG. 17(b) is a block diagram showing an example
11 embodiment of a system for performing real-domain processing
12 of JPEG-coded image data, such that undesired errors in the
13 new transform-coded data are reduced or eliminated in
14 accordance with the present invention;

15 FIG. 18(a) is a block diagram showing an example of a
16 method for performing multiple iterations of the real-domain
17 manipulation of transform-coded data with reduced error,
18 where each iteration is as described in FIG. 16(a) in
19 accordance with the present invention;

20 FIG. 18(b) is a block diagram showing an example of a
21 system for performing multiple iterations of the real-domain
22 manipulation of transform-coded data with reduced error,
23 where each iteration is as described in FIG. 16(b) in
24 accordance with the present invention;

25 FIG. 19(a) shows the same 8x8 block numerical starting
26 point of FIG. 8(c) using the high-precision numbers as input
27 to the forward transform instead of the rounded numbers;

1 FIG. 19(b) shows the same 8x8 block numerical starting
2 point of FIG. 8(d) using the high-precision numbers as input
3 to the forward transform instead of the truncated numbers;

4 FIG. 19(c) shows the same 8x8 block numerical steps as
5 FIG. 8(f); and

6 FIG. 19(d) shows the numerical results when the output
7 of the inverse DCT with rounding, but before clipping, is
8 input to the forward transform followed by quantization.

9 DESCRIPTION OF THE PROBLEM

10 This invention provides methods, systems, and computer
11 products which reduce or eliminate errors introduced by the
12 processing of digital data. Firstly, the source of the
13 error is analyzed and described. This is followed by a
14 presentation of the invention concepts for error reduction
15 and elimination. It is particularly noted that data
16 manipulation and/or processing as employed here-to-before
17 used digital techniques contaminated by the continued
18 introducing of errors by the respective implementation of
19 digital processing. These techniques employed for years are
20 responsible for an inability to maintain original data
21 precision and the continued deterioration of the data
22 representing the phenomenon as more processing is performed.
23 This is particularly detrimental when a process is performed
24 on data which contain errors imparted on the data by
25 previous processes. This results in the continued
26 impairment of the data which thereby becomes less and less
27 useful as more and more processes are performed thereupon.

1 The seriousness of the problem as realized by the
2 inventors of the present invention is described forthwith.
3 It is noted that in the figures presented herein, optional
4 steps are often shown with dashed lines and/or boxes.

5 It is noted that the concepts of the present invention
6 are useful in almost any digital processing technology.
7 However, the subsequent description is mostly related to
8 image data. This is because of the general availability and
9 continued usage of image data compression standards which
10 are employed worldwide. These standards require the
11 introduction into the digital data of the errors to be
12 described and the continued employment and processing of the
13 error contaminated data. These standards basically teach
14 away from the present invention. Thus image technology is a
15 good example for describing the present invention.

16 Figure 1(a) shows an inverse transform method 100.
17 Transform-domain data 'A' 110 are acted on by the inverse
18 transform 120, which produces high-precision real-valued
19 data 130. The high-precision data 130 are converted to
20 integers and clipped 140 to produce integer real-domain data
21 150. In some cases, the integer-valued data are optionally
22 sent to an output device 160.

23 Figure 1(b) shows an inverse transform system 105.
24 Transform-domain data 'A' 115 are acted on by the inverse
25 transformer 125, which produces high-precision real-valued
26 data 135. The high-precision data 135 are input to the
27 integer converter and clipper 145 to produce integer
28 real-domain data 155. In some cases, the integer-valued

2025 RELEASE UNDER E.O. 14176

1 data are optionally input to an output device 165 such as a
2 display monitor, a television set, or an audio player.

3 Figure 2(a) shows a method 200 for decoding
4 transform-coded (i.e. quantized) data. The integer
5 transform-coded data 'B' 210 are inverse quantized 220 (i.e.
6 dequantized) with quantization values as in Equation (2)
7 above. The result of the dequantizing step may then be
8 passed as input to the inverse transform 120, and decoding
9 proceeds as in Figure 1(a).

10 Figure 2(b) shows a system 205 for decoding
11 transform-coded (i.e. quantized) data. The integer
12 transform-coded data 'B' 215 are input to the inverse
13 quantizer 225 with quantization values as in Equation (2)
14 above. The result of the dequantizing step is passed as
15 input to the inverse transformer 125, and decoding proceeds
16 as in Figure 1(b).

17 One aspect of the present invention is concerned with
18 the manipulation of both transform data and transform-coded
19 data. The words "manipulation" and "processing" are used
20 interchangeably herein. Manipulation may be employed in
21 order to achieve many different results. For example, image
22 data must often be processed before printing by scaling
23 and/or rotation. Data from two sources can be merged as is
24 performed in chroma-keying of images or mixing of audio
25 data. Manual manipulation of data is often needed for
26 editing or color correction. Such manipulation of transform
27 data are often performed on the integer real-domain data
28 which results from the transform decoding of Figure 1(a)
29 and/or Figure 2(a).

1 A process for manipulation of transform data 300 is
2 shown in Figure 3. Integer data 150 undergo some form of
3 manipulation 310. If this manipulation 310 does not produce
4 integer output, the manipulated output 340 is again
5 converted to integers and clipped 320. The resulting
6 integer data 330 may be stored, transmitted, and/or
7 optionally sent to an output device 160. Because the stage
8 of clipping and converting to integers 140 is performed
9 before the manipulation which accepts integer input 150, the
10 resulting errors cause the data output from the manipulation
11 340 to contain at least small inaccuracies.

12 It is noted that there is no requirement in the data
13 manipulation processes described above, for the input data
14 to come entirely from one source. For example, many types
15 of data manipulation involve the merging of data from two or
16 more sources. This includes manipulations such as mixing of
17 audio data or merging of images. The processes illustrated
18 in the figures and described generally apply equally well to
19 such types of manipulation. Thus the "input data" used for
20 any of the processes described may in practice come from
21 more than one input source.

22 It is often the case that data after manipulation are
23 to be re-encoded to the transform domain. It is desirable
24 that the process of decoding and re-encoding, when no
25 manipulation is performed on the real-domain data, should be
26 lossless. That is, the data, when the forward transform
27 operation uses the same transform type operation as the
28 inverse transform type of transform operation, should result
29 in exactly the same transform-domain data as was present

1 initially. However, errors are introduced by the converting
 2 to integers and clipping to the allowed range as is
 3 illustrated in Figure 4. Figure 4 shows the integer data
 4 150 used as input to the forward transform device 410, which
 5 accepts integer-valued data as input. The resulting
 6 transform data 'A1' 420 are different from the original
 7 transform data 'A' 110 which were the input to the inverse
 8 transform. This is because the conversion to integers and
 9 the clipping process 140 have introduced errors into the
 10 process. The problem caused by the changes in data after
 11 each iteration, or "generation", of this process is herein
 12 called the "multi-generation problem".

13 The multi-generation problem is also illustrated for
 14 transform-coded data in Figure 5. Here the new
 15 transform-domain data 420 are quantized 510 to produce new
 16 transform-coded data 'B1' 520. It is important to realize
 17 that the quantized data can only change if the errors
 18 produced are larger than half a quantization step:

$$19 \quad Q(\lambda_i + \varepsilon) = Q(\lambda_i) \quad \text{if } |\varepsilon| < 0.5q_i \quad (3)$$

20 where ε is the error produced in this transform coefficient.
 21 This is because each of the λ_i is already a multiple of the
 22 quantization value, since they have been produced by
 23 dequantization as in Equation (2). Thus it is advantageous
 24 to control the errors so that they are sufficiently small.
 25 When the errors are sufficiently small, the new
 26 transform-coded data will be exactly the same as the
 27 original transform-coded data. The maximum possible error
 28 introduced by the conversion to integers by rounding is half
 29 the error introduced by truncating during the conversion.

1 Figure 6 shows a case wherein image manipulation is
2 performed on the data and the resulting modified data are
3 then re-transformed back to the transform domain. The
4 integer data 150 are manipulated as was shown in Figure 3 to
5 produce new integer-valued data 610. These new
6 integer-valued data 610 are used as the input to the forward
7 transform 410 to produce new transform data 'A2' 620. The
8 fact that the process described above without any
9 manipulation produces changes in the transform data 110
10 shows that when manipulation is performed there are
11 undesired changes in the transform data 110 in addition to
12 those which result from the desired manipulation.

13 An example of a method which embodies the process shown
14 in Figure 6, is shown in Figure 7(a). The method 700
15 illustrated performs real-domain manipulation on coded data
16 such as JPEG-coded image data. The coded data 'C' 710 are
17 entropy decoded 720, which is defined for JPEG-coded data in
18 the JPEG standard. The entropy decode step 720 decompresses
19 the data into quantized DCT coefficients. These quantized
20 coefficients are inverse quantized 730 and passed to the
21 inverse transform, which in this system is the
22 two-dimensional 8x8 inverse DCT 740. The resulting
23 real-valued image data are rounded to integers and clipped
24 750 to the allowed range (e.g. [0,255]) to produce
25 integer-valued image data 754 in the allowed range.

26 If it is necessary to show the data before
27 manipulation, for example when the image manipulation is an
28 interactive process, the image can optionally be sent to a
29 display device 758. The image is then manipulated 762 to

1 produce some desired change. If the result of the
2 manipulation is non-integer data then the image data may be
3 converted to integers and clipped to the range e.g. [0,255]
4 768. In this way the image data 772 may again be displayed
5 758. The new real-domain image data 772 are passed to the
6 forward DCT 776 and the resulting DCT coefficients are
7 quantized 780 to produce new quantized DCT coefficients 784.
8 These coefficients 784 are then entropy encoded 788 to
9 produce new coded data 'C1' 792 which are different from the
10 original coded data 'C' 710. Now the new coded data 'C1'
11 792 incorporates not only the desired changes made to the
12 image by the image manipulation 762, but also the errors
13 resulting from the converting and clipping stages 750 and
14 768. It would be advantageous to eliminate or reduce these
15 errors.

16 An example of a system which embodies the process shown
17 in Figure 6, is shown in Figure 7(b). The system 705
18 performs real-domain manipulation on coded data. The coded
19 data 'C' 715 are input to the entropy decoder 725, which is
20 defined for JPEG-coded data in the JPEG standard. The
21 entropy decoder 725 decompresses the data into quantized DCT
22 coefficients. These quantized coefficients are input to the
23 inverse quantizer 735 and the output passed to the inverse
24 transformer, which in this system is the two-dimensional 8x8
25 inverse DCT-er 745. The resulting real-valued image data
26 are rounded to integers and clipped 755 (e.g. to the range
27 [0,255]) to produce integer-valued image data 759 in the
28 allowed range.

1 If it is necessary to show the data before
2 manipulation, for example when the image manipulation is an
3 interactive process, the image can optionally be sent to a
4 display **763**. The image is operated on by a manipulator **767**
5 to produce some desired change. If the result of the
6 manipulation is non-integer data then the image data may be
7 passed to another integer converter and clipper **773**. In
8 this way the image data **777** may again be displayed **763**. The
9 new real-domain image data **777** are passed to the forward
10 DCT-er **781** and the resulting DCT coefficients are input to
11 the quantizer **785** to produce new quantized DCT coefficients
12 **789**. These coefficients **789** are then input to the entropy
13 encoder **793** to produce new coded data 'C1' **797** which are
14 different from the original coded data 'C' **715**. Now the new
15 coded data 'C1' **797** incorporates not only the desired
16 changes made to the image by the image manipulator **767**, but
17 also the errors resulting from the integer converter and
18 clippers **755** and **773**.

19 Figure 8(a) shows the JPEG example luminance
20 quantization matrix **804** for 8x8 DCT luminance blocks. Figure
21 8(b) gives the JPEG example chrominance quantization matrix
22 **814** for 8x8 DCT chrominance blocks. The smallest
23 quantization value in Figure 8(a) is 10. The smallest
24 quantization value in Figure 8(b) is 17. Since the maximum
25 possible error from rounding is 0.5 for each of 64 samples,
26 the largest error in the unquantized forward transform
27 coefficients from conversion to integers by rounding is 4
28 (shown in Figure 8(c)) for JPEG. For the quantization
29 matrices shown in Figures 8(a) and 8(b) this size error is

1 less than half of all of the values and will disappear
 2 during quantization. However, for high quality applications
 3 such as high end printing or digital studio editing, the
 4 quantization matrix values are much smaller. In some cases,
 5 the DC (upper left corner) term is as small as 1 to preserve
 6 maximum quality. Then the rounding errors are significant.

7 The maximum possible error from truncating is just
 8 under 1 for each sample. This almost doubles the error in
 9 the unquantized forward transform coefficients. For the
 10 quantization matrix in Figure 8(a) eight quantization values
 11 are small enough for this error to potentially change the
 12 transform-coded data.

13 A numerical example showing the multi-generation
 14 problem is given in Figure 8(c). In this example the
 15 transform used is the 8x8 DCT as used in the JPEG still
 16 image compression standard. A set of transform-domain
 17 coefficients **822**, of which only one (the constant, or DC,
 18 term) is non-zero, are operated on by the inverse transform
 19 to produce an block of real-domain data **824**. In this case
 20 the data consist of 64 values which are all equal to 128.5.
 21 Note that the JPEG level shift of 128 for 8 bit data has
 22 been applied. The real-domain data are rounded to the
 23 nearest integers **826**, which in this case means that each
 24 value is rounded up to 129. The forward transform is then
 25 applied to produce new transform-domain coefficients **828**.
 26 It can be seen that the resulting new transform coefficients
 27 **828** are significantly different from the initial transform
 28 coefficients **822**. This is a highly undesirable result.

1 This example also applies to transform-coded data if
2 the DC quantization value is set to 1, 2, or 4. Then the
3 transform coefficients **822** would be produced from
4 transform-coded values of 4, 2, or 1 respectively. The
5 quantization of the new transform coefficients **828** would
6 change the resulting DC quantization values to 2, 4, or 8
7 respectively.

8 Another numerical example showing the multi-generation
9 problem is given in Figure 8(d). Again the transform used
10 is the 8x8 DCT as used in the JPEG still image compression
11 standard. A set of transform-domain coefficients **832**, of
12 which only one (the constant, or DC, term) is non-zero, are
13 operated on by the inverse transform to produce a block of
14 real-domain data **834**. In this case the data consist of 64
15 values which are all equal to 128.875. Note that the JPEG
16 level shift of 128 for 8 bit data has been applied. The
17 real-domain data are truncated to the nearest integers **836**,
18 which in this case means that each value is reduced to 128.
19 The forward transform is then applied to produce new
20 transform-domain coefficients **838**. It can be seen that the
21 resulting new transform coefficients **838** are significantly
22 different from the initial transform coefficients **832**. This
23 is a highly undesirable result.

24 Having demonstrated the errors caused by real-domain
25 rounding or truncating when converting to integers, we now
26 show how real-domain clipping can cause errors. Figure 8(e)
27 shows an example of real-domain clipping **850**. This example
28 uses the one-dimensional DCT to illustrate the problem.

1 Figure 8(d) shows a bar chart **854** displaying one block of
2 data consisting of eight samples. The data displayed has
3 only two frequency components: a constant, or DC, component
4 which is indicated by the dotted line; and an alternating,
5 or AC, component which gives an alternating wave pattern
6 symmetrical about the dotted line. The magnitudes of these
7 components, namely the respective DCT coefficients, are
8 high-precision numbers. When quantization is performed,
9 these DCT coefficients are rounded to the nearest
10 quantization level. The data after transform-domain
11 quantization are shown in the bar chart **858**. In the example
12 shown, the DC coefficient has a small quantization value and
13 so quantization does not change the DC level significantly.
14 The AC coefficient shown has a large quantization value and
15 so is changed significantly by quantization. This example
16 shows the AC component almost doubling in magnitude due to
17 quantization. These quantization values reflect, for
18 example, those used when compressing chrominance image data.
19 Thus the data represented after quantization have parts
20 which have negative values. This shows how transform-domain
21 data which, after inverse transforming, give real-domain
22 negative values can be produced by original real-domain data
23 which do not contain negative values.

24 Bar chart **862** shows the data produced from that in
25 chart **858** after real-domain clipping. Those negative parts
26 of the real data have been changed to 0. This results in
27 the DC coefficient of the data increasing and hence leads to
28 error being introduced. Because the quantization value for
29 the DC coefficient is generally small, the error is large

1 enough to cause a change in the quantized data as given in
2 Equation (3).

3 To further illustrate the possibility of error
4 introduced by real-domain clipping, a numerical example 870
5 is shown in Figures 8(f) and 8(g). This example employs the
6 system illustrated in Figure 5. This example uses the
7 two-dimensional 8x8 DCT as used for transform coding of
8 images to illustrate the problem described above. The
9 initial quantized DCT coefficients are shown in matrix 874.
10 All but two of the coefficients are 0; the two non-zero
11 coefficients are the DC coefficient and one high-frequency
12 coefficient. The coefficients, after dequantizing using the
13 quantization matrix shown in Figure 8(a), are shown in
14 matrix 878. When the inverse DCT is performed on these
15 transform data and the level shift of 128 added, real data
16 are produced as shown in matrix 882. The data shown in
17 matrix 882 have already been rounded to integers but have
18 not been clipped to an allowed range. It can be seen that
19 these real data include several negative values. After
20 clipping, the real data 882 produce clipped real data as
21 shown in matrix 886. These data are identical to 882 except
22 that each negative value has been replaced by 0. The
23 forward DCT is then applied to the real-domain data to give
24 new rounded transform data 890. It can be seen that the new
25 transform data are significantly different from the previous
26 transform data 878. When quantization is performed using
27 the quantization matrix shown in Figure 8(a), new
28 transform-coded data 894 are produced. The resulting
29 changes in the transform data are large enough to produce

1 changes in the transform-coded data after quantization.
2 This is a highly undesirable result.

3 In many situations, the process of decoding,
4 manipulation and re-encoding of data needs to be done
5 multiple times. In these situations each iteration of this
6 process is referred to as a "generation". The errors
7 described above, caused by converting to integers and
8 clipping to an allowed range in the real domain, accumulate
9 as multiple iterations are performed and may result in
10 significant degradation of the data. It is realized that the
11 foregoing are only representative examples of errors
12 introduced by rounding (or truncating) and/or clipping.
13 Other examples having more or less error developed are
14 possible.

15 The problem is usually even worse following multiple
16 generations of decoding and re-encoding as shown in Figure
17 9. Initial transform-coded data 'D0' 910 is dequantized and
18 inverse transformed 920, converted to integers and clipped
19 to an allowed range 930 to produce integer-valued
20 real-domain data 940. The real-domain data 940 are passed
21 to the forward transform and quantized 950 to give new
22 transform-coded data 'D1' 960. This whole process is
23 iterated several times, and after some number 'n' of
24 iterations the final transform-coded data 'Dn' 970 is
25 produced. Because of errors in each step the final data
26 'Dn' 970 may be very different from the original data.

27 A case showing the problem significantly worsened due
28 to multiple generations of real-domain manipulation of
29 transform-coded data is shown in Figure 10. In addition to

1 the steps shown in Figure 9, some form of manipulation **310**
 2 is performed on the real-domain data, followed by converting
 3 to integers and clipping **320**. After the forward transform
 4 and quantization, the resulting quantized transform
 5 coefficients **1010** contain some error as in Figure 5. After
 6 'n' generations, the final transform quantized coefficients
 7 **1020** may have quite large undesired errors.

8 DETAILED DESCRIPTION OF THE INVENTION

9 An example embodiment of a method for processing
 10 transform data with reduced error **1100** is illustrated in
 11 Figure 11(a). Transform data 'A' **110** are passed through an
 12 inverse transform **120** to produce high-precision real-domain
 13 data **130**, as in Figure 1(a). If it is necessary to pass the
 14 real-domain data to an output device **160** which takes
 15 integer-valued input, or to generate integer-valued data
 16 before manipulation for any other reason, the steps of
 17 converting to integers and clipping to an allowed range **140**
 18 is done before manipulation without affecting the high-
 19 precision real-domain data. The desired manipulation **1110**
 20 of the real-domain data is performed using a method which
 21 accepts high-precision data as input and produces
 22 high-precision data **1120** as output. This manipulation
 23 method **1110** performs conceptually the same processing on the
 24 data as the manipulation on integers **310** described above in
 25 Figure 3, but operates instead on high-precision data. If
 26 it is necessary to pass the manipulated real-domain data to
 27 an output device **160** which takes integer-valued input, or to

1 generate integer-valued data after manipulation for any
2 other reason, the steps of converting to integers and
3 clipping to an allowed range 140 are done after manipulation
4 without affecting the high precision of the processed data.

5 An example embodiment of a system for processing
6 transform data with reduced error 1105 in accordance with
7 the present invention is illustrated in Figure 11(b).
8 Transform data 'A' 115 are passed through an inverse
9 transformer 125 to produce high-precision real-domain data
10 135, as in Figure 1(b). If it is necessary to pass the
11 real-domain data to an output device 165 which takes
12 integer-valued input, or to generate integer-valued data
13 before manipulation for any other reason, the integer
14 converter and clipper 145 operates before manipulation
15 without affecting the high-precision real-domain data. The
16 manipulator 1115 operates on the real-domain data accepting
17 high-precision data as input and producing high-precision
18 data 1125 as output. This manipulator 1115 performs
19 conceptually the same processing on the data as the
20 manipulation on integers 310 described above in Figure 3,
21 but operates instead on high-precision data. If it is
22 necessary to pass the manipulated real-domain data to an
23 output device 165 which takes integer-valued input, or to
24 generate integer-valued data after manipulation for any
25 other reason, the integer converter and clipper 145 operates
26 after manipulation without affecting the high precision of
27 the processed data.

28 An example of an embodiment of the present invention
29 employing a method for performing inverse transform followed

1 by forward transform steps 1200 is illustrated in Figure
2 12(a). Transform data 'A' 110 are passed through an inverse
3 transform 120 to produce high-precision real-domain data
4 130, as in Figure 1(a). If it is necessary to pass the
5 real-domain data to an output device 160 which takes
6 integer-valued input, or to generate integer-valued data for
7 any other reason, the steps of converting to integers and
8 clipping to an allowed range 140 are done without affecting
9 the high-precision real-domain data. The high-precision
10 data 130 are used as input to the forward transform 1210,
11 which accepts real-valued data as input. The resulting
12 transform data 'A3' 1220 are identical to the original
13 transform data 'A' 110 which were the input to the inverse
14 transform 120 if the forward transform 1210 is the inverse
15 of the inverse transform since the errors from rounding and
16 clipping are not present in the transform data 'A3'. The
17 forward transform 1210 will produce different transform data
18 'A3' 1220 when a different forward transform is used. This
19 allows conversion between transforms without the errors from
20 rounding and clipping being present in the forward transform
21 input.

22 An example of an embodiment of the present invention
23 employing a system with an inverse transformer followed by
24 forward transformer 1205 is illustrated in Figure 12(b).
25 Transform data 'A' 115 are passed through an inverse
26 transformer 125 to produce high-precision real-domain data
27 135, as in Figure 1(b). If it is necessary to pass the
28 real-domain data to an output device 165 which takes
29 integer-valued input, or to generate integer-valued data for

1 any other reason, the integer converter and clipper 145
2 operates without affecting the high-precision real-domain
3 data 135. The high-precision data 135 are used as input to
4 the forward transform 1215, which accepts real-valued data
5 as input. The resulting transform data 'A3' 1225 are
6 identical to the original transform data 'A' 115 which were
7 the input to the inverse transformer 125 if the forward
8 transformer 1215 implements the inverse of the inverse
9 transform since the errors from rounding and clipping are
10 not present in the transform data 'A3'. The forward
11 transformer 1215 will produce different transform data 'A3'
12 1225 when a different forward transformer is used.

13 Figure 13(a) shows a method for performing real-domain
14 manipulation of transform data with reduced error 1300. This
15 method is formed by extending the method 1100 described in
16 Figure 11(a). In this case, the high-precision data 1120 are
17 passed as input to a forward transform 1210 which accepts
18 high-precision data as input, to produce new transform data
19 'A4' 1310 without rounding and/or clipping errors.

20 Figure 13(b) shows a system for performing real-domain
21 manipulation of transform data with reduced error 1305. This
22 method is formed by extending the system 1105 described in
23 Figure 11(b). In this case, the high-precision data 1125 are
24 passed as input to a forward transformer 1215 which accepts
25 high-precision data as input, to produce new transform data
26 'A4' 1315 without rounding and/or clipping errors.

27 A method for performing real-domain manipulation of
28 transform-coded data with reduced error is illustrated in

1 Figure 14(a). Figure 14(a) shows integer transform-coded
2 data 'B' 210 are dequantized 220 and the output passed
3 through an inverse transform 120 to produce high-precision
4 real-domain data 130, as in Figure 2(a). If it is necessary
5 to pass the real-domain data 130 to an output device 160
6 which takes integer-valued input, or to generate
7 integer-valued data 150 before manipulation for any other
8 reason, the steps of converting to integers and clipping to
9 an allowed range 140 are done before manipulation without
10 affecting the high-precision real-domain data 130. The
11 desired manipulation 1110 of the real-domain data is then
12 performed using a method which accepts high-precision data
13 as input and produces high-precision data 1410 as output.
14 This manipulation 1110 performs conceptually the same
15 processing on the data as the manipulation on integers 310
16 described above in Figure 3, but operates instead on
17 high-precision data. If it is necessary to pass the
18 manipulated real-domain data to an output device 160 which
19 takes integer-valued input, or to generate integer-valued
20 data after manipulation for any other reason, the steps of
21 converting to integers and clipping to an allowed range 140
22 are done after manipulation 1110 without affecting the high
23 precision of the processed data 1410.

24 A system for performing real-domain manipulation of
25 transform-coded data with reduced error is illustrated in
26 Figure 14(b). Figure 14(b) shows integer transform-coded
27 data 'B' 215 input to an inverse quantizer 225 and passed
28 through an inverse transformer 125 to produce high-precision
29 real-domain data 135, as in Figure 2(b). If it is necessary

1 to pass the real-domain data **135** to an output device **165**
2 which takes integer-valued input, or to generate
3 integer-valued data **155** before manipulation for any other
4 reason, the integer converter and clipper **145** operates on
5 the data before manipulation without affecting the
6 high-precision real-domain data **135**. The desired
7 manipulation of the real-domain data is then performed using
8 a manipulator **1115** which accepts high-precision data as
9 input and produces high-precision data **1415** as output. This
10 manipulator **1115** performs conceptually the same processing
11 on the data as the manipulation on integers **310** described
12 above in Figure 3, but operates instead on high-precision
13 data. If it is necessary to pass the manipulated
14 real-domain data to an output device **165** which takes
15 integer-valued input, or to generate integer-valued data
16 after manipulation for any other reason, the integer
17 converter and clipper **145** operates on the non-integer data
18 **1415** after manipulation **1115** without affecting the high
19 precision of the processed data **1415**.

20 An example embodiment of a method for real-domain
21 conversion of transform-coded data **1500** is shown in Figure
22 15(a). The high-precision data **130** are used as input to the
23 forward transform **1210**, which accepts real-valued data as
24 input. The output of the forward transform **1210** is quantized
25 **1510**. Depending upon the desired system implementation, the
26 forward transform operation **1210** may employ a different
27 transform than that used in the inverse transform operation
28 **120**. For example, the inverse transform **120** may use the
29 inverse DCT transform whereas the forward transform **1210** may

1 use the Fourier transform. The resulting integer
2 transform-coded data 'B2' 1520 are identical to the original
3 integer transform-coded data 'B' 210 which were the input to
4 the inverse quantize step 220 if the forward transform
5 operation 1210 is the inverse of the inverse transform
6 operation 120 and the quantization values used in the
7 inverse quantization step 220 and the quantization step 1510
8 are identical. It is noted that the forward transform 1210
9 will produce different integer transform-coded data 'B2'
10 when a different forward transform is used. Similarly, use
11 of different quantization values in the inverse quantization
12 220 and quantization 1510 also produces different integer
13 transform-coded data 1520. This method thus allows
14 conversion between transforms and quantization matrices
15 without the errors from rounding and clipping being present
16 in the forward transform 1210 input 130.

17 The conversion between quantization matrices may be for
18 coarser or finer quantization. For converting data from the
19 JPEG international standard to the MPEG international
20 standard, the quantization is likely to be coarser. The
21 higher quality JPEG independent images are needed during the
22 editing process. The coarser, more compressible, MPEG
23 images are used to achieve the desired bandwidth objectives.
24 On the other hand, in recompressing JPEG images after
25 significant hand editing, the quantization is likely to be
26 finer in order to preserve the changes.

27 An example embodiment of a system for real-domain
28 conversion of transform-coded data 1505 in accordance with
29 the present invention is shown in Figure 15(b). The

1 high-precision data 135 are used as input to the forward
2 transformer 1215, which accepts real-valued data as input.
3 The output of the forward transformer 1215 is input to the
4 quantizer 1515. Depending upon the desired system
5 implementation, the forward transformer 1215 may produce a
6 different transform than that used in the inverse
7 transformer 125. For example, the inverse transformer 125
8 may use the inverse DCT transform whereas the forward
9 transformer 1215 may use the Fourier transform. The
10 resulting integer transform-coded data 'B2' 1525 are
11 identical to the original integer transform-coded data 'B'
12 215 which was the input to the inverse quantizer 225 if the
13 forward transformer 1215 produces the inverse of the inverse
14 transformer 125 and the quantization values used in the
15 inverse quantizer 225 and the quantizer 1515 are identical.
16 It is noted that the forward transformer 1215 will produce
17 different integer transform-coded data 'B2' when a different
18 forward transform is produced. Similarly, use of different
19 quantization values in the inverse quantizer 225 and
20 quantizer 1515 also produces different integer
21 transform-coded data 1525. This system thus allows
22 conversion between transforms and quantization matrices
23 without the errors from rounding and clipping being present
24 in the forward transformer 1215 input 135.

25 A method for performing real-domain manipulation of
26 transform-coded data with reduced error 1600 is formed by
27 extending the method 1400 described in Figure 14(a) as is
28 illustrated in Figure 16(a). The high-precision data 1410
29 are passed as input to a forward transform 1210 which

1 accepts high-precision data as input. The output values from
2 the forward transform are quantized 1510 to produce new
3 transform-coded data 'B3' 1610.

4 A system for performing real-domain manipulation of
5 transform-coded data with reduced error 1605 is formed by
6 extending the method 1405 described in Figure 14(b) as is
7 illustrated in Figure 16(b). The high-precision data 1415
8 are passed as input to a forward transformer 1215 which
9 accepts high-precision data as input. The output values from
10 the forward transformer are input to the quantizer 1515 to
11 produce new transform-coded data 'B3' 1615.

12 An example embodiment of a method for real-domain
13 manipulation of transform-coded data with reduced error 1700
14 is shown in Figure 17(a). The chosen embodiment is a method
15 for real-domain manipulation of coded images, which are
16 transform-coded using the DCT. Coded data 'C' 710 are
17 decoded by a lossless entropy decode step 720 to produce
18 quantized DCT coefficients. These coefficients are
19 dequantized 730 and passed through an inverse DCT 740 to
20 produce high-precision real-domain data 1710. If it is
21 necessary to pass the image before manipulation to a display
22 device 758 which takes integer-valued input, or to produce
23 integer-valued data 754 before manipulation for any other
24 reason, the steps of converting to integers and clipping to
25 an allowed range 750 are performed before manipulation 1720
26 without affecting the high-precision real-domain image data
27 1710. The desired manipulation 1720 of the image is then
28 performed using a method which accepts high-precision data
29 as input and produces high-precision data 1730 as output.

1 If it is necessary to pass the manipulated image data to a
2 display 758 which takes integer-valued input, or to generate
3 integer-valued image data 1750 after manipulation for any
4 other reason, the steps of converting to integers and
5 clipping to an allowed range 1740 are performed after
6 manipulation 1720 without affecting the high precision of
7 the processed image data 1730. The high-precision image
8 data 1730 are passed as input to a forward DCT 1760 which
9 accepts high-precision data as input. The output values
10 from the forward transform 1760 are quantized 780 to produce
11 new integer DCT coefficients 1770. These coefficients 1770
12 are encoded by a lossless entropy encode step 788 to produce
13 new coded data 'C2' 1780. If the forward and inverse
14 transforms and the manipulation system are sufficiently
15 accurate so that the error they introduce is less than half
16 a quantization step, as described in Equation (3) given
17 above, no error at all is introduced to the DCT
18 coefficients.

19 An example invention embodiment of a system for
20 real-domain manipulation of transform-coded data with
21 reduced error 1705 is shown in Figure 17(b). The chosen
22 embodiment is to implement a method for real-domain
23 manipulation of coded images such as JPEG-coded images,
24 which are transform-coded using the DCT. Coded data 'C' 715
25 are decoded by a lossless entropy decoder 725 to produce
26 quantized DCT coefficients. These coefficients are sent to
27 a inverse quantizer 735 and then passed through an inverse
28 DCT-er 745 to produce high-precision real-domain data 1715.
29 If it is necessary to pass the image before manipulation to

1 a display device 763 which takes integer-valued input, or to
2 produce integer-valued data 759 before manipulation for any
3 other reason, the integer converter and clipper 755 produces
4 integer-valued data in the allowed range before manipulation
5 1725 without affecting the high-precision real-domain image
6 data 1715. The manipulator 1725 which performs the desired
7 manipulation of the image accepts high-precision data as
8 input and produces high-precision data 1735 as output. If
9 it is necessary to pass the manipulated image data to a
10 display 763 which takes integer-valued input, or to generate
11 integer-valued image data 1755 after manipulation for any
12 other reason, the optional integer converter and clipper
13 1745 produces integer-valued data 1755 after the operation
14 of the manipulator 1725 without affecting the high precision
15 of the processed image data 1735. The high-precision image
16 data 1735 are passed as input to a forward DCT-er 1765 which
17 accepts high-precision data as input. The output values
18 from the forward DCT-er 1765 are sent to the quantizer 785
19 to produce new integer DCT coefficients 1775. These
20 coefficients 1775 are encoded by a lossless entropy encoder
21 793 to produce new coded data 'C2' 1785. If the forward and
22 inverse transforms and the manipulation system are
23 sufficiently accurate so that the error they introduce for
24 each coefficient is less than half a quantization step, as
25 described in Equation (3) given above, no additional error
26 is introduced to the DCT coefficients.

27 A method for performing real-domain manipulations of
28 transform-coded data with reduced error in multiple steps
29 1800, alternating the manipulation steps with forward

1 transforming and quantizing steps and inverse transform and
2 quantizing steps, is illustrated in Figure 18(a). In general
3 each manipulation may perform another operation on the data.
4 For example for digital studio editing, the first
5 manipulation might color correct the image. The second
6 manipulation might merge the color corrected image with a
7 background using the chroma-keying method. The third
8 manipulation might add highlights to the image. The fourth
9 manipulation might crop the image to convert from the 16:9
10 width to height aspect ratio of movies to the 4:3 aspect
11 ratio of television. For the printing of images the first
12 manipulation might rotate the image 90 degrees to orient the
13 image with the printing direction. The second manipulation
14 might merge several independent images into one composite
15 image. A third manipulation might do a color conversion.

16 As shown in Figure 18(a) transform-coded data 'D0' 910
17 are dequantized and passed through an inverse transform 920
18 to produce high-precision real-domain data 1810. If it is
19 necessary to produce integer-valued data for any reason, the
20 high-precision data 1810 may be converted to integers and
21 clipped to an allowed range 1820 without affecting the high
22 precision of the real-domain data 1810. The desired
23 manipulation 1110 of the real-domain data is then performed
24 using a method which accepts high-precision data 1810 as
25 input and produces high-precision data 1840 as output. If
26 it is desired to produce an integer-valued of this output
27 data, the high-precision data 1810 may be converted to
28 integers and clipped to an allowed range 1830 without
29 affecting the high precision of the output data. The

1 high-precision output data are passed as input to a forward
2 transformer and quantizer 1850 to produce new
3 transform-coded data 'F1' 1860. The process of inverse
4 quantizing and inverse transforming, manipulation and
5 forward transforming and quantizing may be repeated multiple
6 times with the manipulation 1870 being different upon each
7 iteration. After multiple steps, final transform-coded data
8 'Fn' 1880 are produced with rounding and/or clipping errors
9 reduced or eliminated. Outputs resulting from any of the
10 convert to integer and clip steps may be sent to an output
11 device 1890 with or without a multiplexor.

12 An example invention embodiment of a system for
13 performing real-domain manipulations of transform-coded data
14 with reduced error in multiple stages 1805, alternating the
15 operation of a manipulator with the operation of a forward
16 transformer and quantizer and the operation of an inverse
17 quantizer and inverse transformer, is illustrated in Figure
18 18(b). Transform-coded data 'D0' 1815 are fed to an inverse
19 quantizer and inverse transformer 1819 to produce
20 high-precision real-domain data 1823. If it is necessary to
21 produce integer-valued data for any reason, the
22 high-precision data 1823 may be operated on by the integer
23 converter and clipper 1827 without affecting the high
24 precision of the real-domain data 1823. The manipulator
25 1115 then operates on the real-domain data 1823 to produce
26 the desired manipulation and produces high-precision data
27 1845 as output. If it is desired to produce integer-values
28 of this output data, the high-precision data 1845 may be
29 input to an integer converter and clipper 1835 without

1 affecting the high precision of the output data. The
 2 high-precision output data are passed as input to a forward
 3 transformer and quantizer **1855** to produce new
 4 transform-coded data 'F1' **1865**. The steps of inverse
 5 quantizing and inverse transforming, manipulation and
 6 forward transforming and quantizing may be repeated multiple
 7 times with the manipulator **1875** being different upon each
 8 iteration. After multiple iterations, final transform-coded
 9 data 'Fn' **1885** are produced with real-domain rounding and/or
 10 clipping errors reduced or eliminated. In a particular
 11 embodiment the output from any or all of the integer
 12 converter and clipper modules is fed to the output device
 13 **1895**. For coded image data the output device may be a
 14 display or television set. For coded audio data the output
 15 device may be a player and/or recorder.

16 A numerical example showing how the present invention
 17 solves one aspect of the multi-generation problem is given
 18 in Figure 19(a). A set of transform-domain coefficients
 19 **822**, of which only one (the constant, or DC, term) is
 20 non-zero, are operated on by the inverse transform to
 21 produce an block of real-domain data **824**. In this case the
 22 data consist of 64 values which are all equal to 128.5.
 23 Note that the JPEG level shift of 128 for 8 bit data has
 24 been applied. The forward transform is then applied to
 25 produce new transform-domain coefficients **1910**. It can be
 26 seen that the new transform coefficients **1910** are identical
 27 to the initial transform coefficients **822**. This is due to
 28 the rounding error not being present in the data sent to the
 29 forward DCT.

2025 RELEASE UNDER E.O. 14176

1 Another numerical example showing how the present
 2 invention solves another aspect of the multi-generation
 3 problem is given in Figure 19(b). A set of transform-domain
 4 coefficients **832**, of which only one (the constant, or DC,
 5 term) is non-zero, are operated on by the inverse transform
 6 to produce an block of real-domain data **834**. In this case
 7 the data consist of 64 values which are all equal to
 8 128.875. Note that the JPEG level shift of 128 for 8 bit
 9 data has been applied. The forward transform is then applied
 10 to produce new transform-domain coefficients **1938**. It can
 11 be seen that the new transform coefficients **1938** are
 12 identical to the initial transform coefficients **832**. This
 13 is due to the truncation error not being present in the data
 14 sent to the forward DCT.

15 Having demonstrated how using the high-precision
 16 numbers removes the errors caused by real-domain rounding or
 17 truncating, we now show how real-domain clipping errors are
 18 also avoided. The same numerical starting point and first
 19 three steps used in Figure 8(f) are shown in Figure 19(c).
 20 The initial quantized DCT coefficients are shown in matrix
 21 **874**. All but two of the coefficients are 0; the two
 22 non-zero coefficients are the DC coefficient and one
 23 high-frequency coefficient. The coefficients after
 24 dequantizing are shown in matrix **878**. The quantization
 25 matrix used is shown in Figure 8(a). When the inverse DCT
 26 is performed on these transform data, real data are produced
 27 as shown in matrix **882**. The data shown in matrix **882** have
 28 already been rounded to integers but have not been clipped
 29 to an allowed range.

1 Figure 19(d) shows the results of the forward DCT
2 applied to the real-domain data to give new rounded
3 transform data **1944**. When quantization is performed, new
4 transform-coded data **1948** are produced. In this example,
5 the changes in the transform data are not large enough to
6 produce changes in the transform-coded data after
7 quantization.

8 Examples of the manipulation between generations
9 include merging two or more transform-coded data sets. For
10 transform-coded image data sets, the merging may be needed
11 because multiple small images need to be collected into one
12 bigger picture. Fan-folded advertising brochures typically
13 are composed of multiple individual pictures. Today's
14 highest end laser printers print more than one page at a
15 time. In such cases, the images generally do not overlap,
16 but may not have the same quantization, positioning relative
17 to the reference grid such as the 8x8 block structure for
18 JPEG DCTs, or orientation. By composing the final picture
19 in the real domain, standard processes can be used for each
20 subimage. Then the composite image can be re-compressed for
21 eventual decompression for on-the-fly printing.

22 Similarly, digital editing can include many special
23 effects requiring several independent manipulations
24 performed serially. Digital movies often use the
25 fade-in/fade-out special effect to perform a smooth
26 transition between two key scenes. Such special effects may
27 follow independent processing of each scene. Thus, multiple
28 generations of decompression and recompression are often
29 needed in the editing to produce the composite of the
30 special effects.

1 Chroma-keying involves two independent video data
2 streams. In one video stream the background has been
3 captured. In the other video stream the foreground, often
4 composed of action involving live actors, has been filmed
5 against a blank single color such as a deep blue or black
6 background. Then the blank pixels in the foreground image
7 are replaced with pixels from the background video. Since
8 the pixels are being mixed at a single-pixel level, the
9 images need to be combined in the real domain. The errors
10 introduced by converting to integers and clipping are highly
11 undesirable for such digital studio applications. These
12 errors are reduced or eliminated by implementing the present
13 invention.

14 Another application example for use of the present
15 invention is in the high-end digital graphics market which
16 uses digital images with sometimes more than 100 megapixels.
17 Glossy advertising brochures and the large photographic
18 trade show booth backdrops are just two examples of the use
19 of such high quality digital imagery. High-quality lossy
20 JPEG compression are sometimes used to keep the transmission
21 and storage costs down. As such images are decompressed and
22 recompressed to allow changes and modifications such as
23 adding highlights, correcting colors, adding or changing
24 text and image cropping, unintentional changes are a problem
25 that is solved with the use of the concepts of the present
26 invention.

27 The above examples for the concepts of the present
28 invention are usual for image and video transform data. The
29 wide use of the Internet has shown the value of JPEG and
30 MPEG compressed image data. When JPEG images are to be

1 printed, then manipulations such as a change of scale or a
2 change of orientation may be required. In addition, a
3 transformation to another color space followed by
4 recompression will allow the print-ready versions of the
5 image to be stored. Use of the present invention overcomes
6 the problem inherent in propagating the errors from the
7 rounding and clipping.

8 Audio coded data also needs to be decompressed, mixed
9 with special sound effects, merged with other audio data,
10 edited and processed in the real domain with reduced errors.
11 Similar implementations are performed for other industrial,
12 commercial, and military applications of digital processing
13 employing a transform and an inverse transform of data
14 representing a phenomenon when the data is stored in the
15 transform domain. These are thus other representative
16 applications wherein use of the present invention is highly
17 advantageous.

18 It is further noted that this invention may also be
19 provided as an apparatus or a computer product. For
20 example, it may be implemented as an article of manufacture
21 comprising a computer usable medium having computer readable
22 program code means embodied therein for causing a computer
23 to perform the methods of the present invention.

24 It is noted that although the description of the
25 invention is made for particular arrangements of steps, the
26 intent and concept of the present invention are suitable and
27 applicable to other arrangements. It will be clear to those
28 skilled in the art that other modifications to the disclosed
29 embodiments can be effected without departing from the

1 spirit and scope of the invention.

2025-01-01 10:00:00